Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER Computers and Mathematics with Applications 49 (2005) 1081–1099
An International Journal
computers & mathematics with applications
www.elsevier.com/locate/camwa

Starting Algorithms for a Class of RK Methods for Index-2 DAEs

I. HIGUERAS AND T. ROLDÁN*
Departamento de Matemática e Informática
Universidad Pública de Navarra
31006 Pamplona, Spain
<higueras><teo>@unavarra.es

(Received and accepted September 2004)

Abstract—When semiexplicit differential-algebraic equations are solved with implicit Runge-Kutta methods (RK), the computational effort is dominated by the cost of solving the nonlinear systems, and therefore it is important to have good starting values to begin the iterations. For semiexplicit index-2 DAEs, starting algorithms without additional cost for RK methods with regular matrix coefficient were studied in a previous paper. However, the regularity condition on the matrix coefficient excludes some interesting methods like Lobatto IIIa and ESDIRK methods. In this paper, we study starting algorithms, without additional computational cost, for a class of Runge-Kutta methods in the case of index-2 DAEs. © 2005 Elsevier Ltd. All rights reserved.

Keywords—Starting algorithms, Differential-algebraic equations, Lobatto IIIA methods, ESDIRK methods.

1. INTRODUCTION

We consider semiexplicit index-2 differential algebraic systems of the form

$$\begin{aligned} y' &= f(y,z), \qquad y(x_0) = y_0, \\ 0 &= g(y), \qquad z(x_0) = z_0, \end{aligned} \tag{1}$$

where $f : \mathbb{R}^l \times \mathbb{R}^m \longrightarrow \mathbb{R}^l$ and $g : \mathbb{R}^l \longrightarrow \mathbb{R}^m$ are sufficiently smooth functions, and the matrix $g_y f_z$ is invertible in a neighborhood of the solution of (1). Furthermore, we assume that the initial values are consistent, i.e., they satisfy the algebraic equation $g(y_0) = 0$ and the hidden constraint $g_y(y_0)f(y_0, z_0) = 0$.

If we consider an s-stage Runge-Kutta method $(\mathcal{A}, b)$ to solve (1), a standard assumption is the matrix $\mathcal{A}$ to be regular. Nevertheless, we can also use methods with singular matrices of the form

$$\begin{array}{c|cc} 0 & 0 & 0^t \\ \bar{c} & a & \bar{\mathcal{A}} \\ \hline & b_1 & \bar{b}^t \end{array} \tag{2}$$

Research supported by the Ministerio de Ciencia y Tecnología, Project BFM2001-2188.
We would like to thank the anonymous referees for their valuable suggestions and detailed comments which have improved the paper.
*Author to whom all correspondence should be addressed.

0898-1221/05/$ - see front matter Typeset by $\mathcal{AMS}$-TEX
doi:10.1016/j.camwa.2004.09.006

where $a \in \mathbb{R}^{s-1}$ and $\bar{A}$ is an $(s-1) \times (s-1)$ regular matrix [1]. Furthermore, we assume that conditions $C(1)$ and $B(1)$ hold, i.e.,

$$a + \bar{A}\bar{e} = \bar{c}, \tag{3}$$

$$b_1 + \bar{b}^t \bar{e} = 1, \tag{4}$$

where $\bar{e} = (1, \ldots, 1)^t \in \mathbb{R}^{s-1}$. Moreover, in order to have $R(\infty)$ bounded, with $R(z)$ the stability function of the method, we impose

$$b_1 - \bar{b}^t \bar{A}^{-1} a = 0. \tag{5}$$

In this way, (3),(4) imply

$$\bar{b}^t \bar{A}^{-1} \bar{c} = 1. \tag{6}$$

For these methods, the first internal stages are $Y_{n+1,1} = y_n$, $Z_{n+1,1} = z_n$, and the rest of the stages are given by the nonlinear system

$$\bar{Y}_{n+1} = \bar{e} \otimes y_n + ha \otimes f(y_n, z_n) + h\left(\bar{A} \otimes I_l\right) f\left(\bar{Y}_{n+1}, \bar{Z}_{n+1}\right), \tag{7}$$

$$0 = g\left(\bar{Y}_{n+1}\right). \tag{8}$$

We have used the notation $\bar{Y}_{n+1} = (Y_{n+1,2}^t, \ldots, Y_{n+1,s}^t)^t \in \mathbb{R}^{l(s-1)}$, $\bar{Z}_{n+1} = (Z_{n+1,2}^t, \ldots, Z_{n+1,s}^t)^t \in \mathbb{R}^{m(s-1)}$; $f(\bar{Y}_{n+1}, \bar{Z}_{n+1}) \in \mathbb{R}^{l(s-1)}$ is the vector $[f(Y_{n+1,2}, Z_{n+1,2})^t, \ldots, f(Y_{n+1,s}, Z_{n+1,s})^t]^t$ and in an analogous way for $g(\bar{Y}_{n+1})$. The symbol $\otimes$ denotes the Kronecker product.

As the matrix $\bar{A}$ is regular, system (7),(8) can be solved for $\bar{Y}_{n+1}$ and $\bar{Z}_{n+1}$. Once these values have been computed, with condition (5), we obtain

$$y_{n+1} = R(\infty) y_n + \left(\bar{b}^t \bar{A}^{-1} \otimes I_l\right) \bar{Y}_{n+1},$$

and similarly, we can compute

$$z_{n+1} = R(\infty) z_n + \left(\bar{b}^t \bar{A}^{-1} \otimes I_m\right) \bar{Z}_{n+1}.$$

If the method is stiffly accurate, i.e., $a_{si} = b_i$, $i = 1, \ldots, s$, we simply obtain

$$y_{n+1} = \bar{Y}_{n+1,s}, \qquad z_{n+1} = \bar{Z}_{n+1,s}.$$

Observe that in this case the numerical solution satisfies $g(y_{n+1}) = 0$. If the method is not stiffly accurate, the numerical solution must be projected onto the constraint $g(y) = 0$ (see [1,2]). Examples of stiffly accurate methods of the form (2) are Lobatto IIIA methods, SDIRK methods in [3], ESDIRK methods considered [4,5], and mono-implicit Runge-Kutta methods in [6,7].

In each step, we have to obtain the internal stage vectors $\bar{Y}_{n+1}$ and $\bar{Z}_{n+1}$ through the resolution of the nonlinear system (7),(8). In the iterative scheme to solve the nonlinear system, we need starting values $(\bar{Y}_{n+1}^{(0)}, \bar{Z}_{n+1}^{(0)})$ as accurate as possible, because in other case, the number of iterations in each step may be too high or even worse, the convergence may fail.

In [8] and [9] a type of initializers for index-1 DAEs was studied and in [8,10] they were extended to the case of index-2 and index-3 DAEs. For the index-1 case, the coefficient matrix $\mathcal{A}$ is not assumed to be regular, and thus, the study made covers methods of the type (2). For example in [8,9] the coefficients of the starting algorithms for the Lobatto IIIA methods with two and three stages were given. For index-2 and index-3 DAEs, the order conditions given in [8,10] involve the inverse of $\mathcal{A}$ and consequently those results are not longer valid.

In this paper, we study initializers to obtain starting values for the internal stages when methods of form (2) are used. In each step these starting values will be obtained using the information from the previous step. We are going to assume that we have just given a step $x_{n-1} \xrightarrow{h} x_n$

from the consistent initial values (y_{n-1}, z_{n-1}), we have calculated the numerical solution (y_n, z_n) at x_n, as well as the internal stages $(\bar{Y}_n, \bar{Z}_n)$, and we are about to give another step $x_n \xrightarrow{rh} x_{n+1}$ to compute the numerical solution (y_{n+1}, z_{n+1}). To achieve this, we have to solve the nonlinear system (7),(8) but now with step hr instead of h in order to consider the most general case of variable step. We propose the following type of starting algorithms:

$$\bar{Y}_{n+1}^{(0)} = \bar{b}_0 \otimes y_{n-1} + \left(\bar{B} \otimes I_l\right) \bar{Y}_n, \tag{9}$$

$$\bar{Z}_{n+1}^{(0)} = \bar{c}_0 \otimes z_{n-1} + \left(\bar{C} \otimes I_m\right) \bar{Z}_n. \tag{10}$$

The vectors $\bar{b}_0$, $\bar{c}_0 \in \mathbb{R}^{s-1}$, and the $(s-1) \times (s-1)$ matrices $\bar{B}$ and $\bar{C}$ will be determined by imposing some order conditions that will be defined later. Notice that the type of initializers considered do not have additional computational cost because the internal stages have been calculated in the previous step. Notice too, that we cannot use this kind of initializers in the first step of the integration.

The rest of the paper is organized as follows. Section 2 begins with a review of the work done in [8,10] for index-2 DAEs. The results obtained in [8,10] are transferred to (2) embedding this method into one with regular coefficient matrix. In Section 3, we show how to construct starting algorithms for some concrete methods. We have also included a brief study about the stability of the predictors considered in Section 4. In Section 5, we give some numerical results to illustrate the performance of the starting algorithms studied in this paper. Some conclusions are given is Section 6.

2. STARTING ALGORITHMS

Given an s-stages Runge-Kutta method $(\mathcal{A}, b)$ with $\mathcal{A}$ regular, the starting algorithms considered in [10] are of the form

$$Y_{n+1}^{(0)} = b_0 \otimes y_{n-1} + (B \otimes I_l) Y_n, \tag{11}$$

$$Z_{n+1}^{(0)} = c_0 \otimes z_{n-1} + (C \otimes I_m) Z_n, \tag{12}$$

where b_0, $c_0 \in \mathbb{R}^s$, and B and C are $s \times s$ matrices which have to be determined.

We say that the starting formula (11),(12) has order (r_y, r_z) if these are the largest integers which satisfy

$$\left\| Y_{n+1}^{(0)} - Y_{n+1} \right\| = \mathcal{O}\left(h^{r_y+1}\right), \qquad \left\| Z_{n+1}^{(0)} - Z_{n+1} \right\| = \mathcal{O}\left(h^{r_z+1}\right).$$

The vectors b_0, c_0 and the matrices B, C are determined so that these algorithms achieve the maximum possible order in each variable. To obtain this, we need the series both for the initializers $(Y_{n+1}^{(0)}, Z_{n+1}^{(0)})$ and for the internal stages (Y_{n+1}, Z_{n+1}).

In [10], the $DA2$-series theory (see [11,12]) is used. $DAT2 = DAT2_y \cup DAT2_z$ denotes the set of rooted trees with two types of vertex, meagre and fat; the expression $[t_1, \ldots, t_\mu, u_1, \ldots, u_\nu]_y$ denotes the tree which is obtained by joining the roots of $t_1, \ldots, t_\mu, u_1, \ldots, u_\nu$ to a meagre vertex; $[t_1, \ldots, t_\mu]_z$ denotes the tree obtained by joining the roots of $t_1, \ldots, t_\mu$ to a fat vertex, provided $t_1 \neq [u]_y$ if $\mu = 1$. τ denotes the tree consisting of a single meagre vertex. The order of a tree $t \in DAT2$, denoted by $\rho(t)$, is the difference between the number of meagre and fat vertices of that tree. Given two vectors $u, v \in \mathbb{R}^s$, $u \bullet v$ denotes the product component by component.

The internal stages of the Runge-Kutta method [11, Theorem 5.7] can be written as $DA2$-series in terms of the coefficients $\Phi_y(t)$ and $\Phi_z(u)$ which are defined by

$$\begin{aligned}
&\Phi_y(\emptyset_y) = e, \qquad \Phi_z(\emptyset_z) = e, \qquad \Phi_y(\tau) = c, \\
&\Phi_y(t) = \rho(t) \mathcal{A}[\Phi_y(t_1) \bullet \cdots \bullet \Phi_y(t_\mu) \bullet \Phi_z(u_1) \bullet \cdots \bullet \Phi_z(u_\nu)], \\
&\quad \text{if } t = [t_1, \ldots, t_\mu, u_1, \ldots, u_\nu]_y \in DAT2_y, \\
&\Phi_z(u) = \frac{1}{\rho(u)+1} \mathcal{A}^{-1}[\Phi_y(t_1) \bullet \cdots \bullet \Phi_y(t_\mu)], \\
&\quad \text{if } u = [t_1, \ldots, t_\mu]_z \in DAT2_z.
\end{aligned} \tag{13}$$

Observe that these coefficients $\Phi_y(t)$ and $\Phi_z(u)$ are related to the coefficients $\phi_y(t)$ and $\phi_z(u)$ defined in [11] by the formula

$$\Phi_y(t) = \gamma(t)\mathcal{A}\phi_y(t), \qquad \Phi_z(u) = \gamma(u)\phi_z(u).$$

In the construction of the initializers, the functions $\bar{\Phi}_y, \bar{\Phi}_z \in \mathbb{R}^s$ defined by

$$\begin{aligned}
&\bar{\Phi}_y(\emptyset_y) = e, \qquad \bar{\Phi}_z(\emptyset_z) = e, \qquad \bar{\Phi}_y(\tau) = e + rc,\\
&\bar{\Phi}_y(t) = \rho(t)\left[eb^t \prod_{i=1}^{\mu} \Phi_y(t_i) \bullet \prod_{j=1}^{\nu} \Phi_z(u_j) + r\mathcal{A} \prod_{i=1}^{\mu} \bar{\Phi}_y(t_i) \bullet \prod_{j=1}^{\nu} \bar{\Phi}_z(u_j)\right],\\
&\quad \text{if } t = [t_1, \ldots, t_\mu, u_1, \ldots, u_\nu]_y \in DAT2_y,\\
&\bar{\Phi}_z(u) = \frac{1}{r(\rho(u)+1)}\mathcal{A}^{-1}\left[-eb^t\mathcal{A}^{-1}\prod_{i=1}^{\mu}\Phi_y(t_i) + \prod_{i=1}^{\mu}\bar{\Phi}_y(t_i)\right],\\
&\quad \text{if } u = [t_1, \ldots, t_\mu]_z \in DAT2_z,
\end{aligned} \tag{14}$$

where the coefficients Φ_y and Φ_z are the ones defined in (13), are also needed.

In [10] it is proved that (11) reaches order r_y for the differential variable if this is the largest integer which satisfies

$$\begin{aligned}
&b_0 + Be = e,\\
&B\Phi(t) = \bar{\Phi}(t), \qquad \forall t \in DAT2_y, \quad \text{with } 1 \le \rho(t) \le r_y,
\end{aligned} \tag{15}$$

and (12) reaches order r_z for the algebraic variable if this is the largest integer which satisfies

$$\begin{aligned}
&c_0 + Ce = e,\\
&C\Phi(u) = \bar{\Phi}(u), \qquad \forall u \in DAT2_z, \quad \text{with } 1 \le \rho(u) \le r_z.
\end{aligned} \tag{16}$$

Tables with the order conditions for the differential component up to order 4, and for the algebraic one up to order 3, are given in [8] (in [10] up to orders 3 and 2, respectively). For the differential variable there is a condition of order 1, two of order 2, six of order 3, and 21 of order 4, whereas for the algebraic one there are two of order 1, six of order 2, and 21 of order 3. These conditions can be reduced considerably in the particular case that the method satisfies some simplifying conditions. The following ones are usual in the theory of Runge-Kutta methods:

$$B(p): \qquad b^t c^{k-1} = \frac{1}{k}, \qquad k = 1, \ldots, p,$$

$$C(q): \qquad \mathcal{A}c^{k-1} = \frac{c^k}{k}, \qquad k = 1, \ldots, q,$$

$$D(r): \qquad \left(b \bullet c^{k-1}\right)^t \mathcal{A} = \frac{1}{k}\left[b^t - \left(b \bullet c^k\right)^t\right], \qquad k = 1, \ldots, r.$$

With these conditions there are many trees that give the same order condition, and thus, they can be omitted. In [10] it is proved that the order conditions to get order $q+1$ for the differential component (cf. (15)) when $B(p)$ and $C(q)$, $p \ge q \ge 1$, hold true are

$$\begin{aligned}
&Bc^k = (e + rc)^k, \qquad k = 1, \ldots, q+1,\\
&B\mathcal{A}c^q = eb^t c^q + r\mathcal{A}(e + rc)^q.
\end{aligned}$$

For the algebraic component, when $B(p)$ and $C(q)$, $p \ge q \ge 1$, hold true, the conditions to obtain order q (cf. (16)) are

$$\begin{aligned}
&Cc^k = (e + rc)^k, \qquad k = 1, \ldots, q,\\
&C\mathcal{A}^{-1}c^{q+1} = \frac{1}{r}\mathcal{A}^{-1}\left[-eb^t\mathcal{A}^{-1}c^{q+1} + (e + rc)^{q+1}\right].
\end{aligned}$$

It is also proved in [10] that for nonconfluent methods, i.e., $c_i \neq c_j$ for $i \neq j$, under the assumptions $B(p)$ and $C(q)$ with $p \geq q \geq s-1$, the maximum order is q for the differential component and $s-1$ for the algebraic one. Recall that this result requires high-stage order.

Another way of finding the starting algorithm is by interpolation and extrapolation. In this case, in [10] it is proved that for nonconfluent methods with coefficients $c_i \neq 0$, $i = 1, \ldots, s$ satisfying $C(q)$, the starting algorithms obtained by interpolation and extrapolation have order q for the differential variable and order $q-1$ for the algebraic one. Thus for methods with low-order stage, the starting algorithms obtained by interpolation and extrapolation give poor results.

Recall once more that the study done in [10] is valid only for regular matrices $\mathcal{A}$, and thus, it is not directly applicable to the methods considered in this paper (2). In order to make use of those results, we embed method (2) into the ε-method

$$\begin{array}{c|cc} \varepsilon & \varepsilon & 0^t \\ \bar{c} & a & \bar{A} \\ \hline & b_1 & \bar{b}^t \end{array} = \begin{array}{c|c} c_\varepsilon & \mathcal{A}_\varepsilon \\ \hline & b^t \end{array}.$$

If $\varepsilon \neq 0$, the coefficient matrix is regular, and thus, we can try to apply the results in [8,10]. Although in a different way, the idea of embedding the method in another one has already been used for example in [13].

The internal stages $Y_{n,\varepsilon}$, $Z_{n,\varepsilon}$ of this numerical method converge to $(y_{n-1}, \bar{Y}_n)$ and $(z_{n-1}, \bar{Z}_n)$ when ε tends to zero. The idea is to construct starting algorithms like (11),(12) for the ε-method

$$\begin{pmatrix} Y^{(0)}_{n+1,1,\varepsilon} \\ \bar{Y}^{(0)}_{n+1,\varepsilon} \end{pmatrix} = \begin{pmatrix} \beta_{01,\varepsilon} \\ \bar{\beta}_{0,\varepsilon} \end{pmatrix} \otimes y_{n-1} + \left(\begin{pmatrix} b_{11,\varepsilon} & b^t_{f,\varepsilon} \\ b_{c,\varepsilon} & \bar{B}_\varepsilon \end{pmatrix} \otimes I_l \right) \begin{pmatrix} Y_{n,1,\varepsilon} \\ \bar{Y}_{n,\varepsilon} \end{pmatrix},$$

$$\begin{pmatrix} Z^{(0)}_{n+1,1,\varepsilon} \\ \bar{Z}^{(0)}_{n+1,\varepsilon} \end{pmatrix} = \begin{pmatrix} \gamma_{01,\varepsilon} \\ \bar{\gamma}_{0,\varepsilon} \end{pmatrix} \otimes z_{n-1} + \left(\begin{pmatrix} c_{11,\varepsilon} & c^t_{f,\varepsilon} \\ c_{c,\varepsilon} & \bar{C}_\varepsilon \end{pmatrix} \otimes I_m \right) \begin{pmatrix} Z_{n,1,\varepsilon} \\ \bar{Z}_{n,\varepsilon} \end{pmatrix}$$

by using the results in [8,10], and take the limit when ε tends to zero to obtain

$$\begin{aligned} Y^{(0)}_{n+1,1} &= (\beta_{01} + b_{11}) y_{n-1} + \left(b^t_f \otimes I_l\right) \bar{Y}_n, \\ \bar{Y}^{(0)}_{n+1} &= \left(\bar{\beta}_0 + b_c\right) \otimes y_{n-1} + \left(\bar{B} \otimes I_l\right) \bar{Y}_n, \qquad (17) \\ Z^{(0)}_{n+1,1} &= (\gamma_{01} + c_{11}) z_{n-1} + \left(c^t_f \otimes I_m\right) \bar{Z}_n, \\ \bar{Z}^{(0)}_{n+1} &= \left(\bar{\gamma}_0 + c_c\right) \otimes z_{n-1} + \left(\bar{C} \otimes I_m\right) \bar{Z}_n, \qquad (18) \end{aligned}$$

where we have assumed that $Y_{n,1} = y_{n-1}$ and $Z_{n,1} = z_{n-1}$. Recall that we need starting algorithms only for $\bar{Y}_{n+1}$ and $\bar{Z}_{n+1}$, and thus, we have to construct only the matrices $\bar{B}$, $\bar{C}$ and the vectors $\bar{\beta}_0 + b_c$ and $\bar{\gamma}_0 + c_c$. Observe that the starting algorithms (17) and (18) are the form (9),(10).

As it has been pointed out above, the starting algorithm for the ε-method will be constructed imposing the order conditions obtained in [10]. Taking into account how the starting algorithms are constructed, we simply have to prove that the expressions involved in the order conditions (15),(16), namely $\Phi_{y,\varepsilon}$, $\Phi_{z,\varepsilon}$, $\bar{\Phi}_{y,\varepsilon}$, and $\bar{\Phi}_{z,\varepsilon}$, are bounded when ε tends to zero. Recall that the matrix $\mathcal{A}_\varepsilon^{-1}$, which contains the term $1/\varepsilon$, is involved in the definition of these functions.

For the coefficients

$$\begin{array}{c|c} \bar{c} & \bar{A} \end{array}$$

we define the functions $\psi_y(t) : DAT2_y \to \mathbb{R}^{s-1}$ and $\psi_z(u) : DAT2_z \to \mathbb{R}^{s-1}$ recursively as

$$\begin{aligned}
&\psi_y(\tau) = \bar{c},\\
&\psi_y(t) = \rho(t)\bar{A}[\psi_y(t_1) \bullet \cdots \bullet \psi_y(t_\mu) \bullet \psi_z(u_1) \bullet \cdots \bullet \psi_z(u_\nu)],\\
&\quad \text{if } t = [t_1, \ldots, t_\mu, u_1, \ldots, u_\nu]_y \in DAT2_y,\\
&\psi_z(u) = \frac{1}{\rho(u)+1}\bar{A}^{-1}[\psi_y(t_1) \bullet \cdots \bullet \psi_y(t_\mu)],\\
&\quad \text{if } u = [t_1, \ldots, t_\mu]_z \in DAT2_z.
\end{aligned} \tag{19}$$

The following result gives the expression of the functions $\Phi_{y,\varepsilon}$ and $\Phi_{z,\varepsilon}$ for the ε-method.

LEMMA 1. *For the ε-method, the functions $\Phi_{y,\varepsilon}$ and $\Phi_{z,\varepsilon}$ satisfy*

$$\Phi_{*,\varepsilon}(t) = \begin{pmatrix} \mathcal{O}\left(\varepsilon^{\rho(t)}\right) \\ \psi_*(t) + \mathcal{O}(\varepsilon) \end{pmatrix}, \qquad \forall t \in DAT2, \quad \rho(t) \geq 1.$$

PROOF. An induction process on $\rho(t)$ gives the desired result. There is only a tree with order one in $DAT2_y$, namely $t = \tau$. For this tree $\Phi_y(\tau) = c$, and thus,

$$\Phi_{y,\varepsilon}(\tau) = \begin{pmatrix} \varepsilon \\ \bar{c} \end{pmatrix} = \begin{pmatrix} \varepsilon \\ \psi_y(\tau) \end{pmatrix}.$$

There are two trees with order 1 in $DAT2_z$, namely $u_{1,1} = [\tau, \tau]_z$ and $u_{1,2} = [[\tau]_y]_z$. We have $\Phi_z(u_{1,1}) = (1/2)\mathcal{A}^{-1}c^2$ and $\Phi_z(u_{1,2}) = c$. For the ε-method, we have

$$\Phi_{z,\varepsilon}(u_{1,1}) = \frac{1}{2}\mathcal{A}_\varepsilon^{-1}c_\varepsilon^2 = \frac{1}{2}\begin{pmatrix} \varepsilon \\ \bar{A}^{-1}\bar{c}^2 - \varepsilon\bar{A}^{-1}a \end{pmatrix} = \begin{pmatrix} \frac{1}{2}\varepsilon \\ \psi(u_{1,1}) - \frac{1}{2}\varepsilon\bar{A}^{-1}a \end{pmatrix},$$

$$\Phi_{z,\varepsilon}(u_{1,2}) = \begin{pmatrix} \varepsilon \\ \bar{c} \end{pmatrix} = \begin{pmatrix} \varepsilon \\ \psi_z(u_{1,2}) \end{pmatrix}.$$

We assume now that the result is true for any tree in DAT_y, with $\rho(t) \leq p$. We consider the tree $t = [t_1, \ldots, t_\mu, u_1, \ldots, u_\nu]_y \in DAT2_y$ with $\rho(t) = p+1$. For each tree t_i, $i = 1, \ldots, \mu$, and u_k, $k = 1, \ldots, \nu$, we have $\rho(t_i) \leq p$ and $\rho(u_k) \leq p$, and thus, from (13) and (19) we immediately obtain the desired expression. We consider now a tree $u = [t_1, \ldots, t_\mu]_z \in DAT_z$ with $\rho(u) = p+1$. If $\mu > 1$, then $\rho(t_i) \leq p+1$, $i = 1, \ldots, \mu$. As for these trees, we have already proved that the result is true, from (19) we obtain the desired result. It remains to consider the tree $u = [t]_z$, with $t = [t_1, \ldots, t_\mu, u_1, \ldots, u_\nu]_y$, $(\mu, \nu) \neq (0, 1)$ and $\rho(t) = p+2$. Observe that now $\rho(t_i) \leq p+1$, $\rho(u_j) \leq p$ and for these trees we have already proved that the result holds true. From (13) and (19), we obtain the desired result. ∎

With the above lemma, without any proof we can state the following result.

PROPOSITION 2. *The functions $\Phi_{y,\varepsilon}(t)$ and $\Phi_{z,\varepsilon}(u)$ are bounded when ε tends to zero.*

We find now the expressions for $\bar{\Phi}_{y,\varepsilon}$, $\bar{\Phi}_{z,\varepsilon}$.

LEMMA 3. *If method (2) satisfies*

$$\bar{b}^t\bar{A}^{-1}\psi_y(t) = 1, \qquad \forall t \in DAT2_y, \quad 1 \leq \rho(t) \leq p,$$

and those trees $t \in DAT2_y$ of order $\rho(t) = p+1$ which are of the form $t = [u]_y$, with $u \in DAT2_z$, then the functions $\bar{\Phi}_{y,\varepsilon}(t)$ and $\bar{\Phi}_{z,\varepsilon}(u)$ for the ε-method satisfy

$$\bar{\Phi}_{y,\varepsilon}(t) = \begin{pmatrix} 1 + \mathcal{O}(\varepsilon) \\ \mathcal{O}(1) \end{pmatrix}, \qquad \forall t \in DAT2_y, \quad \rho(t) \leq p, \tag{20}$$

$$\bar{\Phi}_{z,\varepsilon}(u) = \mathcal{O}(1), \qquad \forall u \in DAT2_z, \quad \rho(u) \leq p. \tag{21}$$

PROOF. We follow an induction process on $\rho(t)$. For the unique tree $\tau \in DAT2_y$ with order 1, we have $\bar{\Phi}_y(\tau) = e + rc$ and

$$\bar{\Phi}_{y,\varepsilon}(\tau) = \begin{pmatrix} 1 + r\varepsilon \\ \bar{e} + r\bar{c} \end{pmatrix}.$$

Thus, expression (20) holds trivially. For the two trees with order 1 in $DAT2_z$, we have $\bar{\Phi}_z(u_{1,1}) = (1/2r)\mathcal{A}^{-1}[-eb^t\mathcal{A}^{-1}c^2 + (e + rc)^2]$ and $\bar{\Phi}_z(u_{1,2}) = e + rc$. For the ε-method, we have

$$\begin{aligned}
\bar{\Phi}_{z,\varepsilon}(u_{1,1}) &= \frac{1}{2r}\mathcal{A}_\varepsilon^{-1}\left[-eb^t\mathcal{A}_\varepsilon^{-1}c_\varepsilon^2 + (e + rc_\varepsilon)^2\right] \\
&= \frac{1}{2r}\begin{pmatrix} \dfrac{\left(1 - \bar{b}^t\bar{\mathcal{A}}^{-1}\bar{c}^2\right)}{\varepsilon} + 2r + \varepsilon r^2 \\ \left(-\dfrac{\left(1 - \bar{b}^t\bar{\mathcal{A}}^{-1}\bar{c}^2\right)}{\varepsilon} - 2r - \varepsilon r^2\right)\bar{\mathcal{A}}^{-1}a + \mathcal{O}(1) \end{pmatrix}, \\
\bar{\Phi}_{z,\varepsilon}(u_{1,2}) &= \begin{pmatrix} 1 + r\varepsilon \\ \bar{e} + r\bar{c} \end{pmatrix}.
\end{aligned}$$

Thus, if $\bar{b}^t\bar{\mathcal{A}}^{-1}\bar{c}^2 = 1$, as $\psi([u_{1,1}]_y) = \bar{c}^2$, we obtain expression (21) for $u_{1,1}$ and $u_{1,2}$.

We assume now that (20),(21) are true for any tree in $DAT2_y$, with $\rho(t) \leq p$. We consider the tree $t = [t_1, \ldots, t_\mu, u_1, \ldots, u_\nu]_y \in DAT2_y$ with $\rho(t) = p + 1$. For each tree t_i, $i = 1, \ldots, \mu$, and u_k, $k = 1, \ldots, \nu$, we have $\rho(t_i) \leq p$, and $\rho(u_k) \leq p$. From (14), we obtain

$$\begin{aligned}
\bar{\Phi}_{y,\varepsilon}(t) &= \rho(t)\left[e\left(b_1, \bar{b}^t\right)\begin{pmatrix} \mathcal{O}\left(\varepsilon^{(\rho(t_1)+\cdots+\rho(u_\nu))}\right) \\ \prod_{i=1}^{\mu}\psi_y(t_i) \bullet \prod_{j=1}^{\nu}\psi_z(u_j) + \mathcal{O}(\varepsilon) \end{pmatrix} + r\begin{pmatrix} \varepsilon & 0 \\ a & \bar{\mathcal{A}} \end{pmatrix}\begin{pmatrix} \mathcal{O}(1) \\ \mathcal{O}(1) \end{pmatrix}\right] \\
&= \begin{pmatrix} \bar{b}^t\bar{\mathcal{A}}^{-1}\psi_y(t) + \mathcal{O}(\varepsilon) \\ \mathcal{O}(1) \end{pmatrix},
\end{aligned}$$

and thus, (20) holds.

We consider now a tree in DAT_z with $\rho(u) = p + 1$ of the form $u = [t_1, \ldots, t_\mu]_z$. If $\mu > 1$, then $\rho(t_i) \leq p + 1$, $i = 1, \ldots, \mu$. For these trees, we have already proved that expression (20) holds. From (14), we obtain

$$\begin{aligned}
\bar{\Phi}_{z,\varepsilon}(u) &= \frac{1}{r(\rho(u)+1)}\mathcal{A}_\varepsilon^{-1}\left[-eb^t\mathcal{A}_\varepsilon^{-1}\begin{pmatrix} \varepsilon^{(\rho(u)+1)} \\ \prod_{i=1}^{\mu}\psi_y(t_i) + \mathcal{O}(\varepsilon) \end{pmatrix} + \begin{pmatrix} 1 + \mathcal{O}(\varepsilon) \\ \mathcal{O}(1) \end{pmatrix}\right] \\
&= \frac{1}{r\left(\rho(u)+1\right)}\mathcal{A}_\varepsilon^{-1}\begin{pmatrix} -\bar{b}^t\bar{\mathcal{A}}^{-1}\prod_{i=1}^{\mu}\psi_y(t_i) + 1 + \mathcal{O}(\varepsilon) \\ \mathcal{O}(1) \end{pmatrix} = \mathcal{O}(1).
\end{aligned}$$

Observe that $\prod_{i=1}^{\mu}\psi_y(t_i) = \psi_y([u]_y)$ and $\rho([u]_y) = p + 2$.

Finally, we consider the tree $u = [t]_z$, with $t = [t_1, \ldots, t_\mu, u_1, \ldots, u_\nu]_y$, with $(\mu, \nu) \neq (0, 1)$ and $\rho(t) = p + 2$. Observe that now $\rho(t_i) \leq p + 1$, $\rho(u_j) \leq p$, and for these trees (20) holds. From (14), we obtain

$$\begin{aligned}
\bar{\Phi}_{z,\varepsilon}(u) &= \frac{1}{r(\rho(u)+1)}\mathcal{A}_\varepsilon^{-1}\left[-eb^t\mathcal{A}_\varepsilon^{-1}\begin{pmatrix} \varepsilon^{(\rho(u)+1)} \\ \psi_y(t) + \mathcal{O}(\varepsilon) \end{pmatrix} + \begin{pmatrix} 1 + \mathcal{O}(\varepsilon) \\ \mathcal{O}(1) \end{pmatrix}\right] \\
&= \frac{1}{r\left(\rho(u)+1\right)}\mathcal{A}_\varepsilon^{-1}\begin{pmatrix} -\bar{b}^t\bar{\mathcal{A}}^{-1}\psi_y(t) + 1 + \mathcal{O}(\varepsilon) \\ \mathcal{O}(1) \end{pmatrix}.
\end{aligned}$$

Observe that $\psi_y(t) = \psi_y([u]_y)$. ∎

With the previous lemma, the following result is straightforward. Recall that we assume (6).

PROPOSITION 4. *If method (2) satisfies*

$$\bar{b}^t \bar{\mathcal{A}}^{-1} \psi_y(t) = 1, \qquad \forall t \in DAT2_y, \quad 1 \leq \rho(t) \leq p, \tag{22}$$

and those trees $t \in DAT2_y$ of order $\rho(t) = p+1$ which are of the form $t = [u]_y$, with $u \in DAT2_z$, then

1. *The function $\bar{\Phi}_{y,\varepsilon}(t)$, $t \in DAT2_y$, $\rho(t) \leq p+2$, is bounded when ε tends to zero.*
2. *The function $\bar{\Phi}_{z,\varepsilon}(u)$, $u \in DAT2_z$, $\rho(u) \leq p$, is bounded when ε tends to zero.*

Thus given (2), we can consider the ε-method and use the tables given in [8,10] to construct the starting algorithms for the ε-method. In this way, when ε tends to zero we obtain starting algorithms for method (2). The process is valid, i.e., the limits involved exit, if conditions in Proposition 4 hold.

As it has been pointed out previously, in the particular case that the method satisfies some simplifying assumptions, the order conditions for the starting algorithms are simplified considerably. In general, the simplifying conditions are not transferred from the original method (2) to the ε-method. For example, the ε-method only satisfies $C(1)$ with independence of the $C(q)$ condition satisfied by (2). This fact is not a drawback because as we will take the limit when ε tends to zero, it is enough to consider the simplifying assumptions in the limit case. We give the following definition.

DEFINITION 5. *For the ε-method, we define the simplifying assumptions as follows:*

$$B_\varepsilon(p): \qquad \lim_{\varepsilon \to 0} \left(b^t c_\varepsilon^{k-1} - \frac{1}{k} \right) = 0, \qquad k = 1, \ldots, p,$$

$$C_\varepsilon(q): \qquad \lim_{\varepsilon \to 0} \left(\mathcal{A}_\varepsilon c_\varepsilon^{k-1} - \frac{c_\varepsilon^k}{k} \right) = 0, \qquad k = 1, \ldots, q,$$

$$D_\varepsilon(r): \qquad \lim_{\varepsilon \to 0} \left(\left(b \bullet c_\varepsilon^{k-1}\right)^t \mathcal{A}_\varepsilon - \frac{1}{k} \left[b^t - \left(b \bullet c_\varepsilon^k\right)^t \right] \right) = 0, \qquad k = 1, \ldots, r.$$

For the class of methods (2) considered in this paper, the simplifying assumptions are:

$$B(p): \qquad b_1 + \bar{b}^t \bar{e} = 1, \qquad k = 1,$$

$$\bar{b}^t \bar{c}^{k-1} = \frac{1}{k}, \qquad k = 2, \ldots, p;$$

$$C(q): \qquad a + \bar{\mathcal{A}} \bar{e} = \bar{c}, \qquad k = 1,$$

$$\bar{\mathcal{A}} \bar{c}^{k-1} = \frac{\bar{c}^k}{k}, \qquad k = 2, \ldots, q;$$

$$D(r): \qquad \left(\bar{b} \bullet \bar{c}^{k-1}\right)^t a = \frac{1}{k} b_1,$$

$$\left(\bar{b} \bullet \bar{c}^{k-1}\right)^t \bar{\mathcal{A}} = \frac{1}{k} \left[\bar{b}^t - \left(\bar{b} \bullet \bar{c}^k\right)^t \right], \qquad k = 1, \ldots, r.$$

The simplifying conditions of the original method and the ε-method are related as follows.

PROPOSITION 6.

1. *Method (2) satisfies $B(p)$ if and only if the ε-method satisfies $B_\varepsilon(p)$.*
2. *Method (2) satisfies $C(q)$ if and only if the ε-method satisfies $C_\varepsilon(q)$.*
3. *Method (2) satisfies $D(r)$ if and only if the ε-method satisfies $D_\varepsilon(r)$.*

PROOF. For the first case, we compute

$$b^t c_\varepsilon^{k-1} - \frac{1}{k} = b_1 \varepsilon^{k-1} + \bar{b}^t \bar{c}^{k-1} - \frac{1}{k}.$$

Thus, $B_\varepsilon(p)$ trivially holds true if and only if $B(p)$ does. For the other two cases, we only have to compute

$$\mathcal{A}_\varepsilon c_\varepsilon^{k-1} - \frac{1}{k}\begin{pmatrix}\varepsilon^k\\ \bar{c}^k\end{pmatrix} = \begin{pmatrix}\left(1-\frac{1}{k}\right)\varepsilon^k\\ a\varepsilon^{k-1} + \bar{\mathcal{A}}\bar{c}^{k-1} - \frac{1}{k}\bar{c}^k\end{pmatrix}$$

and

$$\begin{aligned}&\left(b \bullet c_\varepsilon^{k-1}\right)^t \mathcal{A}_\varepsilon - \frac{1}{k}\left[b^t - \left(b \bullet c_\varepsilon^k\right)^t\right]\\ &= \left(b_1\left(1+\frac{1}{k}\right)\varepsilon^k + \left(\bar{b} \bullet \bar{c}^{k-1}\right)^t a - \frac{1}{k}b_1,\right.\\ &\quad \left.\left(\bar{b} \bullet \bar{c}^{k-1}\right)^t \bar{\mathcal{A}} - \frac{1}{k}\left[\bar{b}^t - \left(\bar{b} \bullet \bar{c}^k\right)^t\right]\right).\end{aligned}$$

∎

Thus, we can use for the ε-method the same simplifying conditions as for method (2). In practice it means that we can use the reduced tables given in [8,10].

PROPOSITION 7. *Let us consider the RK method (2) whose coefficients $(\mathcal{A}, b)$ satisfy the simplifying assumptions $B(p)$ and $C(q)$ with $p \geq q \geq 1$. Then*

1. *The function $\bar{\Phi}_{y,\varepsilon}(t)$ is bounded when ε tends to zero for all $t \in DAT2_y$, with $\rho(t) \leq q+1$.*
2. *The function $\bar{\Phi}_{y,\varepsilon}(t)$ is bounded when ε tends to zero for all $t \in DAT2_y$, with $\rho(t) \leq q+2$ if and only if*
$$\bar{b}^t \bar{\mathcal{A}}^{-1} \bar{c}^{q+1} = 1. \tag{23}$$
3. *The function $\bar{\Phi}_{z,\varepsilon}(u)$ is bounded when ε tends to zero for all $u \in DAT2_z$, with $\rho(u) \leq q$, if and only if (23) holds.*
4. *The function $\bar{\Phi}_{z,\varepsilon}(u)$ is bounded when ε tends to zero for all $u \in DAT2_z$, with $\rho(u) \leq q+1$, if and only if condition (23) together with conditions*
$$\bar{b}^t \bar{\mathcal{A}}^{-1} \bar{c}^{q+2} = 1, \tag{24}$$
$$\bar{b}^t \bar{\mathcal{A}}^{-1}\left(\bar{c} \bullet \bar{\mathcal{A}}\bar{c}^q\right) = \bar{b}^t \bar{c}^q \tag{25}$$
hold.

PROOF. We will use the reduced tables given in [8,10] together with condition (22) in Proposition 4.

(i) Observe that if $p \geq q \geq 1$, up to order q, the trees in $DAT2_y$ are reduced to those of height one, i.e., trees of the form $[\tau, \ldots, \tau]_y$. Condition (22) in this case is $\bar{b}^t \bar{\mathcal{A}}^{-1} \bar{c}^k = 1$, $k = 2, \ldots, q$, which is satisfied because $B(p)$ and $C(q)$ hold.

(ii) In this case, we have to consider the trees of order $q+1$ in $DAT2_y$. These are reduced to two ones, one of the form $[\tau, \ldots, \tau]_y$ and the other of the form $[[\tau, \ldots, \tau]_z]_y$. Condition (22) only affects to the second one, and for this tree, this condition is (23).

(iii) This case is analogous to the second one (the same condition is needed), but now for the algebraic variable. By Proposition 4 the function $\bar{\Phi}_{z,\varepsilon}(u)$ is bounded up to $\rho(u) \leq q$.

(iv) We have to study the trees of order $q+2$ in $DAT2_y$. Condition (22) only affects those trees of the form $[[\tau, \ldots, \tau]_z]_y$ and $[[[\tau, \ldots, \tau]_y, \tau]_z]_y$. For the first tree this condition is (24), whereas for the second one the condition is given by (25). ∎

REMARK 1. If the method is stiffly accurate then conditions (23) and (24) hold trivially. ∎

We can impose the order conditions for the ε-method in [10, Tables 2.3 and 2.4] and take the limit when ε tends to zero. In this way, we obtain the order conditions showed in Tables 1 and 2.

Table 1. Order conditions with $B(p)$, $C(q)$, $p \geq q \geq 1$ (differential variable) provided condition (23).

Order	Condition	ODE
0	$\bar{b}_0 + \bar{B}\bar{e} = \bar{e}$	*
1	$\bar{B}\bar{c} = \bar{e} + r\bar{c}$	*
2	$\bar{B}\bar{c}^2 = (\bar{e} + r\bar{c})^2$	*
$\vdots$	$\vdots$	
q	$\bar{B}\bar{c}^q = (\bar{e} + r\bar{c})^q$	*
$q+1$	$\bar{B}\bar{c}^{q+1} = (\bar{e} + r\bar{c})^{q+1}$	
	$\bar{B}\bar{A}\bar{c}^q = \bar{e}\bar{b}^t\bar{c}^q + r\left(a + \bar{A}(\bar{e} + r\bar{c})^q\right)$	*
$q+2$	$\bar{B}\bar{A}\bar{c}^{q+1} = \bar{e}\bar{b}^t\bar{c}^{q+1} + r\left(a + \bar{A}\,(\bar{e} + r\bar{c})^{q+1}\right)$	*
	$\bar{B}\bar{c}^{q+2} = (\bar{e} + r\bar{c})^{q+2}$	
	$\bar{B}\bar{A}\left[\bar{c}\cdot\bar{A}^{-1}\bar{c}^{q+1}\right] = \bar{e}\bar{b}^t\left(\bar{c}\cdot\bar{A}^{-1}\bar{c}^{q+1}\right) + (q+1)ra$	
	$\quad + \bar{A}\left[(\bar{e} + r\bar{c})\cdot\bar{A}^{-1}\left((\bar{e} + r\bar{c})^{q+1} - (q+1)ra - \bar{e}\right)\right]$	
	$\bar{B}\bar{A}^2\bar{c}^q = r^2\bar{A}a + \bar{e}\bar{b}^t\bar{A}\bar{c}^q + r\bar{c}\bar{b}^t\bar{c}^q + r^2\bar{A}^2\,(\bar{e} + r\bar{c})^q$	*
	$\bar{B}\left[\bar{c}\cdot\bar{A}\bar{c}^q\right] = (\bar{e} + r\bar{c})\cdot\left[\bar{e}\bar{b}^t\bar{c}^q + r(a + \bar{A}(\bar{e} + r\bar{c})^q)\right]$	

Table 2. Order conditions with $B(p)$, $C(q)$, $p \geq q \geq 1$ (algebraic variable) provided conditions (24) and (25).

Order	Condition
0	$\bar{c}_0 + \bar{C}\bar{e} = \bar{e}$
1	$\bar{C}\bar{c} = \bar{e} + r\bar{c}$
2	$\bar{C}\bar{c}^2 = (\bar{e} + r\bar{c})^2$
$\vdots$	$\vdots$
$q-1$	$\bar{C}\bar{c}^{q-1} = (\bar{e} + r\bar{c})^{q-1}$
q	$\bar{C}\bar{c}^q = (\bar{e} + r\bar{c})^q$
	$\bar{C}\bar{A}^{-1}\bar{c}^{q+1} = \dfrac{1}{r}\bar{A}^{-1}\left[-(q+1)ra + (\bar{e} + r\bar{c})^{q+1} - \bar{e}\right]$
$q+1$	$\bar{C}\bar{c}^{q+1} = (\bar{e} + r\bar{c})^{q+1}$
	$\bar{C}\bar{A}^{-1}\bar{c}^{q+2} = \dfrac{1}{r}\bar{A}^{-1}\left[-(q+2)ra + (\bar{e} + r\bar{c})^{q+2} - \bar{e}\right]$
	$\bar{C}\bar{A}\bar{c}^q = \bar{e}\bar{b}^t\bar{c}^q + r\left(a + \bar{A}\,(\bar{e} + r\bar{c})^q\right)$
	$\bar{C}\left[\bar{c}\cdot\bar{A}^{-1}\bar{c}^{q+1}\right] = \dfrac{1}{r}\,(\bar{e} + r\bar{c})\cdot\bar{A}^{-1}\left[-(q+1)ra - \bar{e} + (\bar{e} + r\bar{c})^{q+1}\right]$
	$\bar{C}\bar{A}^{-1}\left[\bar{c}\cdot\bar{A}\bar{c}^q\right] = \dfrac{-1}{r}\bar{A}^{-1}\left[ar\left(1 + \bar{b}^t\bar{c}^q\right) + \bar{e}\bar{b}^t\bar{A}^{-1}\left(\bar{A}\bar{c}^q\cdot\bar{c}\right)\right.$
	$\quad \left. - (r\bar{c} + \bar{e})\left(ra + r\bar{A}\,(r\bar{c} + \bar{e})^2 + \bar{e}\bar{b}^t\bar{c}^q\right)\right]$

In Table 1, we have marked (*) the order conditions for singular methods applied to the ODE case.

Next, we give results on the maximum order possible for the starting algorithms following the lines in [9,10].

THEOREM 8. *Let us consider an s-stage IRK method (2) whose coefficients $(\mathcal{A}, b)$ satisfy the simplifying assumptions $B(p)$ and $C(q)$ with $p \geq q \geq s-2$. Then, for the starting methods proposed in (9),(10) in the case of the differential variable y:*

(y1) *for $q = s-2$, if the range of*

$$\begin{pmatrix} \bar{e} & \bar{c} & \cdots & \bar{c}^{s-2} \\ 1 & 0 & \cdots & 0 \end{pmatrix} \tag{26}$$

is maximum, then there is an $(s-1)$-parametric family of starting algorithms with order $s-2$, and there is not any of order $s-1$;

(y2) *for $q=s-1$ and $q=s$, if the method is nonconfluent, there is a unique starting algorithm with order $s-1$. Order s is not possible.*

For the starting methods proposed in (9),(10) in the case of the algebraic variable z:

(z1) *for $q=s-2$, and if $\det(\bar{c},\bar{c}^2,\ldots,\bar{c}^{s-2},\bar{A}^{-1}\bar{c}^{s-1})\neq 0$, then there is a unique starting algorithms with order $s-2$, there is not any of order $s-1$;*

(z2) *for $q=s-1$, if the range of matrix (26) is maximum, then there exists an $(s-1)$-parametric family of starting algorithms with order $s-2$, it is not possible to get order $s-1$;*

(z3) *for $q=s$, if the method is nonconfluent, there is a unique starting algorithm with order $s-1$, there is not any of order s.*

PROOF. The proof can be done either with help of the ε-method or directly, following the lines of the proofs in [8,10]. We follow this second approach because it gives the hint on how to construct the predictors.

We have to compute the $(s-1)\times(s-1)$-matrix $\bar{B}$ and the $(s-1)$-vector $\bar{b}_0$ in (9). For the differential variable, the order conditions up to order $q+1$ are given in Table 1.

(y1) If $q=s-2$, then the $s-1$ order conditions up to order $s-2=q$ are given by

$$(\bar{B}\quad \bar{b}_0)\begin{pmatrix}\bar{e} & \bar{c} & \cdots & \bar{c}^{s-2}\\ 1 & 0 & \cdots & 0\end{pmatrix}=\left(\bar{e},(\bar{e}+r\bar{c}),\ldots,(\bar{e}+r\bar{c})^{s-2}\right). \tag{27}$$

As the range of (26) is maximum $(s-1)$, we can obtain an $(s-1)$-parametric family of order $s-2$. It is not possible to impose the two conditions of order $s-1=q+1$, and thus, order $s-1$ is not possible.

(y2) If $q=s-1$, then the s order conditions up to order $s-1=q$ are given by

$$(\bar{B}\quad \bar{b}_0)\begin{pmatrix}\bar{e} & \bar{c} & \cdots & \bar{c}^{s-1}\\ 1 & 0 & \cdots & 0\end{pmatrix}=\left(\bar{e},(\bar{e}+r\bar{c}),\ldots,(\bar{e}+r\bar{c})^{s-1}\right). \tag{28}$$

Now, by the nonconfluence of the method, there is a unique starting algorithm of order $s-1$. If $q=s$, then the s order conditions up to order $s-1=q-1$ are also given by (28), and thus, we have the same result. In any case, it is possible to impose the conditions of order s.

For the algebraic variable, we have to compute the $(s-1)\times(s-1)$-matrix $\bar{C}$ and the $(s-1)$-vector $\bar{c}_0$ in (9). The order conditions up to order q are given in Table 2.

(z1) If $q=s-2$, then the s order conditions up to order $s-2=q$ are given by

$$(\bar{C}\quad \bar{c}_0)\begin{pmatrix}\bar{e} & \bar{c} & \cdots & \bar{c}^{s-2} & \bar{A}^{-1}\bar{c}^{s-1}\\ 1 & 0 & \cdots & 0 & 0\end{pmatrix}=\left(\bar{e},(\bar{e}+r\bar{c}),\ldots,(\bar{e}+r\bar{c})^{s-2},\bar{\Psi}_z\right)$$

with $\bar{\Psi}_z=(1/r)\bar{A}^{-1}[-(s-1)ra+(\bar{e}+r\bar{c})^{s-1}-\bar{e}]$. Thus, there is a unique starting algorithm of order $s-2$ if $\det(\bar{c},\bar{c}^2,\ldots,\bar{c}^{s-2},\bar{A}^{-1}\bar{c}^{s-1})\neq 0$. It is not possible to impose the five conditions of order $s-1=q+1$, and thus, order $s-1$ is not possible.

(z2) If $q=s-1$, then the $s-1$-order conditions up to order $s-2=q-1$ are given by (27) with $\bar{C}$ and $\bar{c}_0$ instead of $\bar{B}$ and $\bar{b}_0$. Thus there is an $(s-1)$-parametric family of order $s-2$. It is not possible to impose the five conditions of order $s-1=q+1$, and thus, order $s-1$ is not possible.

(z3) If $q=s$, then the s-order conditions up to order $s-1=q-1$ are given by (28) with $\bar{C}$ and $\bar{c}_0$ instead of $\bar{B}$ and $\bar{b}_0$. Thus there is a unique starting algorithm of order $s-1$. It is not possible to impose the two conditions of order q. ∎

REMARK 2. For the differential variable, both in the case $q = s-1$ as in $q = s$, the nonconfluence of the Runge-Kutta method (2) guarantees the regularity of the matrix

$$\begin{pmatrix} \bar{e} & \bar{c} & \cdots & \bar{c}^{s-1} \\ 1 & 0 & \cdots & 0 \end{pmatrix}$$

in (28). If the method is confluent, then it is not possible to achieve order $s-1$. The same situation occurs for the algebraic variable in the case $q = s$. ∎

It has been also pointed out in Section 2 that another way to construct starting algorithms is by interpolation and extrapolation. This way of getting starting values is a particular case of the methods defined in (9),(10). In the following result, we give the order of these starting values.

PROPOSITION 9. *Let us consider a nonconfluent IRK method (2) satisfying $C(q)$. Then $Y^{(0)}_{n+1,i}$, $Z^{(0)}_{n+1,i}$, the initializers by interpolation and extrapolation, have orders $\min\{q, s-1\}$ and $q-1$, respectively.*

PROOF. The Runge-Kutta method is nonconfluent and $c_1 = 0$, then the interpolation error is $\mathcal{O}(h^s)$. We obtain the desired result if we take into account that condition $C(q)$ implies $Y_{n,i} - y(x_{n-1} + c_i h) = \mathcal{O}(h^{q+1})$ and $Z_{n,i} - z(x_{n-1} + c_i h) = \mathcal{O}(h^q)$. ∎

Observe that the order obtained by interpolation and extrapolation can be lower when the stage order q is poor.

In Table 3, we have summarized the results of Theorem 8 for nonconfluent IRK methods (2). Observe that for methods satisfying $C(s-2)$, for the algebraic variable, the optimum starting algorithm may have one order more than the one obtained by interpolation and extrapolation.

Table 3. Order for nonconfluent methods.

	$C(s)$		$C(s-1)$		$C(s-2)$	
	Int.-Ext.	Opt.	Int.-Ext.	Opt.	Int.-Ext.	Opt.
Y	$s-1$	$s-1$	$s-1$	$s-1$	$s-2$	$s-2$ (f)
Z	$s-1$	$s-1$	$s-2$	$s-2$ (f)	$s-3$	**$s-2$**

3. EXAMPLES

In this section, we consider some methods and show how to construct the optimal starting algorithms.

3.1. Lobatto IIIA Methods

These are nonconfluent stiffly accurate methods (2) satisfying $C(s)$ and $B(2s-2)$. Applying Theorem 8, we obtain that there is a unique starting algorithm of order $s-1$ both for the differential and the algebraic components. This optimum initializer coincides with the one obtained by interpolation and extrapolation. For $s = 2$ it is given by

$$Y^{(0)}_{n+1,2} = -r y_{n-1} + (1+r) y_n,$$

and similarly for $Z^{(0)}_{n+1,2}$. For $s = 3$ it is given by

$$Y^{(0)}_{n+1,2} = \frac{r(1+r)}{2} y_{n-1} - r(2+r) Y_{n,2} + \frac{2+3r+r^2}{2} y_n,$$

$$Y^{(0)}_{n+1,3} = \frac{r+2r^2}{2} y_{n-1} - 4r(1+r) Y_{n,2} + \left(1+3r+2r^2\right) y_n.$$

and similarly for $Z^{(0)}_{n+1,2}$.

3.2. Four Stages Method with $B(3)$ and $C(2)$

We consider the family of four stages stiffly accurate methods considered in [3] satisfying $B(3)$ and $C(2)$,

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 2\lambda & \lambda & \lambda & 0 & 0 \\ c_3 & \dfrac{6c_3\lambda - 4\lambda^2 - c_3^2}{4\lambda} & \dfrac{c_3 u_1}{4\lambda} & \lambda & 0 \\ 1 & \dfrac{12u_2\lambda^2 + 6u_3\lambda - u_3}{12c_3\lambda} & \dfrac{6\lambda u_2 + u_3}{12\lambda u_1} & \dfrac{6\lambda^2 - 6\lambda + 1}{3c_3 u_1} & \lambda \end{array} \tag{29}$$

where

$$u_1 = c_3 - 2\lambda, \qquad u_2 = 1 - c_3, \qquad u_3 = 3c_3 - 2,$$

and $c_3 \neq 0, 2\lambda$. The starting values used in [3] are

$$Y_{n+1,i}^{(0)} = \begin{cases} y_n, & \text{if } c_i > \max\{c_j : j = 1, \ldots, i-1\}, \\ \dfrac{[(c_i - c_l)Y_{n,j} - (c_i - c_j)Y_{n,l}]}{(c_j - c_l)}, & \text{if } c_i \in [c_l, c_j],\ c_l \neq c_j,\ j, l < i. \end{cases}$$

This starting algorithm is of form (9) but it only satisfies the consistency condition. Observe that the stepsize ratio r is not included in it.

For this method, we have $C(2)$ and $s = 4$, therefore by Proposition 9 the predictor by interpolation and extrapolation achieves order 2 for the differential variable and order 1 for the algebraic one. However, from Theorem 8, we obtain that it is possible to construct an $(s-1)$-parametric family of starting algorithms of order 2 for the differential variable, and a starting algorithm of order 2 for the algebraic component provided that

$$-c_3 + 3\lambda + 3c_3\lambda + 6\lambda^2 \neq 0. \tag{30}$$

Imposing the conditions up to order 2,

$$\bar{B}\bar{c} = \bar{e} + r\bar{c}, \qquad \bar{B}\bar{c}^2 = (\bar{e} + r\bar{c})^2,$$

we obtain

$$\begin{aligned} \bar{b}_{11} &= \frac{\bar{b}_{13} - \bar{b}_{13}c_3 + (-1 + c_3 - 2\lambda r)(1 + 2\lambda r)}{2(c_3 - 2\lambda)\lambda}, \\ \bar{b}_{12} &= \frac{1 + \bar{b}_{13}(-1 + 2\lambda) + 4\lambda^2(-1 + r)r + \lambda(-2 + 4r)}{c_3(c_3 - 2\lambda)}, \\ \bar{b}_{21} &= \frac{\bar{b}_{23} - (1 + c_3 r)^2 + c_3\left(1 - \bar{b}_{23} + c_3 r\right)}{2(c_3 - 2\lambda)\lambda}, \\ \bar{b}_{22} &= \frac{\bar{b}_{23}(-1 + 2\lambda) - (-1 + 2\lambda - c_3 r)(1 + c_3 r)}{c_3(c_3 - 2\lambda)}, \\ \bar{b}_{31} &= \frac{\bar{b}_{33} - \bar{b}_{33}c_3 + (-1 + c_3 - r)(1 + r)}{2(c_3 - 2\lambda)\lambda}, \\ \bar{b}_{32} &= \frac{1 - \bar{b}_{33} - 2\lambda + 2\bar{b}_{33}\lambda + 2r - 2\lambda r + r^2}{c_3(c_3 - 2\lambda)}. \end{aligned}$$

Now, from the consistency condition $\bar{b}_0 = \bar{e} - \bar{B}\bar{e}$, we have

$$\begin{aligned} \bar{b}_{01} &= \frac{\bar{b}_{13}(-1 + c_3 + 2\lambda - 2c_3\lambda) + (1 + 2\lambda(-1 + r))(1 - c_3 + 2\lambda r)}{2c_3\lambda}, \\ \bar{b}_{02} &= \frac{\bar{b}_{23}(-1 + c_3 + 2\lambda - 2c_3\lambda) + (1 + c_3(-1 + r))(1 - 2\lambda + c_3 r)}{2c_3\lambda}, \\ \bar{b}_{03} &= \frac{\bar{b}_{33}(-1 + c_3 + 2\lambda - 2c_3\lambda) - (-1 + c_3 - r)(1 - 2\lambda + r)}{2c_3\lambda}. \end{aligned}$$

There are still three free parameters, $\bar{b}_{13}$, $\bar{b}_{23}$, and $\bar{b}_{33}$. We can try to impose one of the two conditions of order 3 for the differential variable or we can try to impose order 2 conditions for the algebraic variable. If we proceed in this second way, the starting algorithm will be valid both for the differential and the algebraic variable and it is advantageous from the point of view of the implementation.

The order conditions up to order 2 for the algebraic variable are the same as the ones for the differential variable plus

$$\bar{C}\bar{A}^{-1}\bar{c}^3 = \frac{1}{r}\bar{A}^{-1}\left[-3ra - \bar{e} + (\bar{e}+r\bar{c})^3\right]. \tag{31}$$

Imposing this condition, we determine $\bar{b}_{13}$, $\bar{b}_{23}$, and $\bar{b}_{33}$, and thus, we obtain the unique starting algorithm of order 2 both for the differential and the algebraic variable. Due to their length, we do not present here the coefficients $\bar{b}_{j3}$. Recall that it is possible to impose the order condition (31) if and only if condition (30) holds.

In [3] different values of λ and c_3 are considered. In the examples below, we have considered $\lambda \approx 0.43586652$ to get $R(\infty) = 0$, and $c_3 = 1.153799789$. In this way, Theorem 5.2 in [1] ensures order of convergence 3 for the differential component y and order 2 for the algebraic component z.

3.3. Six Stages Method with $B(4)$ and $C(2)$

We consider the stiffly accurate six stages method considered in [5] satisfying $B(4)$ and $C(2)$,

$$\begin{array}{c|cccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 \\
\frac{83}{250} & \frac{8611}{62500} & -\frac{1743}{31250} & \frac{1}{4} & 0 & 0 & 0 \\
\frac{31}{50} & \frac{5012029}{34652500} & -\frac{654441}{2922500} & \frac{174375}{388108} & \frac{1}{4} & 0 & 0 \\
\frac{17}{20} & \frac{15267082809}{155376265600} & -\frac{71443401}{120774400} & \frac{730878875}{902184768} & \frac{2285395}{8070912} & \frac{1}{4} & 0 \\
1 & \frac{82889}{524892} & 0 & \frac{15625}{83664} & \frac{69875}{102672} & -\frac{2260}{8211} & \frac{1}{4} \\
\hline
 & \frac{82889}{524892} & 0 & \frac{15625}{83664} & \frac{69875}{102672} & -\frac{2260}{8211} & \frac{1}{4}
\end{array} \tag{32}$$

This method has order four for ODEs and DAEs with index 1. For index-2 DAEs (1) it has order 3 for the differential component y and order 2 for the algebraic component z [1, Theorem 5.2]. This method has been proved to be efficient for the solution of the discretized unsteady compressible Navier-Stokes equations [14].

For this method, the starting algorithm obtained by interpolation and extrapolation has order 2 for the differential component and 1 for the algebraic one. However, it is possible to construct an starting algorithm with order 3 for the differential component and 2 for the algebraic one. To do so, we simply have to impose the conditions stated in Tables 1 and 2, namely

$$\begin{aligned}
\bar{B}\bar{c} &= \bar{e} + r\bar{c}, \\
\bar{B}\bar{c}^2 &= (\bar{e}+r\bar{c})^2, \\
\bar{B}\bar{c}^3 &= (\bar{e}+r\bar{c})^3, \\
\bar{B}\bar{A}\bar{c}^2 &= \bar{e}\bar{b}^t\bar{c}^2 + r\left(a + \bar{A}(\bar{e}+r\bar{c})^2\right), \\
\bar{B}\bar{A}^{-1}\bar{c}^3 &= \frac{1}{r}\bar{A}^{-1}\left[-(q+1)ra + (\bar{e}+r\bar{c})^3 - \bar{e}\right].
\end{aligned}$$

Once that $\bar{B}$ has been calculated, we obtain $\bar{b}_0$ from the consistence condition

$$\bar{b}_0 + \bar{B}\bar{e} = \bar{e}.$$

4. STABILITY

In the previous sections, we have studied the order properties of the predictors proposed. In this section, we deal with some aspects on stability of these predictors taking into account the results obtained recently in [15] for ordinary differential equations.

If we integrate the Protero and Robinson's problem

$$y' = \lambda(y - \phi(t)) + \phi'(t), \qquad y(0) = y_0,$$

with a Runge-Kutta method $(\mathcal{A}, b)$ like in [15], the global error at t_n is given by

$$y(t_n) - y_n = \hat{R}(z)\,(\phi(t_{n-1}) - y_{n-1}) + d_h(t_{n-1}),$$

where $\hat{R}(z)$ is the stability function of the Runge-Kutta method. The first term, $\hat{R}(z)(\phi(t_{n-1}) - y_{n-1})$, represents the propagation error from the previous step, whereas the second one, $d_h(t_{n-1})$, is the local error.

It is possible to get a similar result for the difference between the internal stages and the predictors $Y_i - Y_i^{(0)}$, obtaining

$$Y_i - Y_i^{(0)} = \left(\bar{R}_i(z) - \tilde{R}_i(z)\right)(y_{n-1} - \phi(t_{n-1})) + \ell_h(t_{n-1}), \tag{33}$$

where

$$\bar{R}_i(z) = \left(1 + zb^t(I - z\mathcal{A})^{-1}e\right)\left(e_i^t(I - rz\mathcal{A})^{-1}e\right)$$

and

$$\tilde{R}_i(z) = 1 + ze_i^t B\mathcal{A}(I - z\mathcal{A})^{-1}e$$

are the stability functions of Y_i and $Y_i^{(0)}$, respectively (for details see [15]), and the second term $\ell_h(t_{n-1})$ in (33) refers to the order of the predictors already studied in the previous section.

Whereas in the previous section, we have tried to obtain a given order cancelling as many terms as possible in $\ell_h(t_{n-1})$, in this section, we focus on the difference $\bar{R}_i(z) - \tilde{R}_i(z)$. From the stability point of view it is important that this difference remains bounded and hence the sets

$$D_i = \left\{z \in \mathbb{C} \mid \left|\bar{R}_i(z) - \tilde{R}_i(z)\right| \leq 1\right\}, \qquad i = 2, \ldots, s,$$

play an important role. Remember that the first stage is explicit for the considered methods and therefore it is not initialized.

We have obtained the domain D_i for each of the methods (29) and (32) considered in the previous section. In Figure 1, we show D_2 for different values of the step ratio r, namely $r = 1/8, 1/4, 1/2, 1, 2, 4$, for the method (29). We see that the region D_2 increases when the step size ratio r is diminished and decreases when r is increased. We only show the set D_2 because D_3 and D_4 are very similar. For $i = 2$ there is a critical value, namely $r_0 \approx 0.42182$, such that for $r < r_0$ the domain D_2 contains the left complex half-plane. A similar situation holds for D_3 and D_4.

Observe that D_i is very small for big values of r, which could restrict the step size for some problems. That is why the code should foresee moderate values of r. For high values of r, it could be more suitable to use other predictors with better stability properties. For example, we have computed the sets D_i, $i = 2, \ldots, s$, for the trivial predictor and, for any value of r, the domain D_i contains all the left complex half-plane.

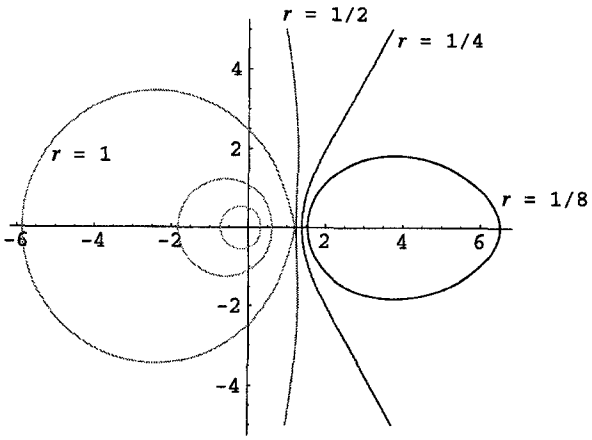

Figure 1. D_2 for Runge-Kutta (29).

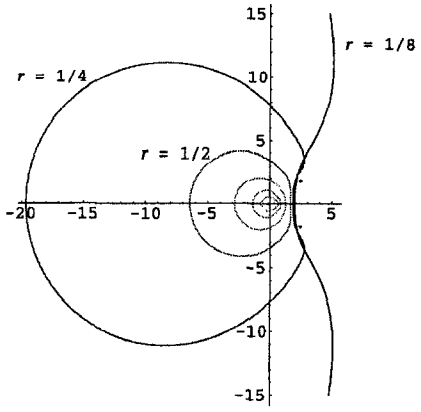

Figure 2. D_5 for Runge-Kutta (32).

This situation is repeated for method (32). In this case, we will only show D_5 because for the other stages the results are quite similar. In Figure 2, we show D_5 for different values or step ratio, namely $r = 1/8, 1/4, 1/2, 1, 2, 4$.

For this method there is also a critical value of the step ratio from which the domain D_i contains the left complex half-plane. For the case $i = 5$ this critical value is $r_0 = 0.125807$.

5. NUMERICAL EXPERIMENTS

In this section, we show the efficiency of the starting algorithms studied in this paper. The results obtained agree with the ones obtained in previous papers [9,10,16] where the use of a high-order starting algorithm improves the performance of the code.

PROBLEM 1. *We consider the index-2 DAE*

$$\begin{aligned} y' &= y^2 + z + \cos t - 1, \\ v' &= y^2 + v^2 - 1 - \sin t, \qquad t \in [1,2], \\ 0 &= y^2 + v^2 - 1, \end{aligned}$$

whose solution is given by

$$y(t) = \sin t, \qquad v(t) = \cos t, \qquad z(t) = \cos^2 t.$$

Table 4. Numerical results for Problem 1.

	Stepsize	$(\mathrm{Tol}_y, \mathrm{Tol}_z)$	Iter./Stage Optimal	Iter./Stage Trivial
Four-stages method (29)	0.02	$(10^{-7}, 10^{-3})$	2.16	4.00
	0.01	$(10^{-7}, 10^{-3})$	2.06	3.82
	0.005	$(10^{-7}, 10^{-4})$	1.21	3.69
	0.0025	$(10^{-7}, 10^{-4})$	1.07	3.15
	0.001	$(10^{-7}, 10^{-6})$	1.18	4.00
Six-stages method (32)	0.02	$(10^{-7}, 10^{-3})$	1.14	3.80
	0.01	$(10^{-7}, 10^{-3})$	1.05	3.37
	0.005	$(10^{-7}, 10^{-4})$	1.04	3.42
	0.0025	$(10^{-7}, 10^{-4})$	1.02	3.10
	0.001	$(10^{-7}, 10^{-6})$	1.03	3.74

The consistent initial conditions have been taken at $t = 1$. We have integrated the problem with constant stepsize. The criteria to stop the simplified Newton iterations is

$$\frac{\left\|Y^{(k)} - Y^{(k-1)}\right\|}{\left\|Y^{(k)}\right\|} < \mathrm{TOL}_y, \qquad \frac{\left\|Z^{(k)} - Z^{(k-1)}\right\|}{\left\|Z^{(k)}\right\|} < \mathrm{TOL}_z\,. \tag{34}$$

In Table 4, we show the average number of iterations per stage for the methods considered in Sections 3.2 and 3.3. Remember that for these type of methods the stages are solved sequentially. It can be seen that the use of a high-order starting algorithm decreases the computational cost.

PROBLEM 2. *We consider the index-2 DAE*

$$\begin{aligned}
y_1' &= y_5 + 2y_1z_3 - 2(y_3 - y_1)z_4,\\
y_2' &= y_6 + 2y_2z_3 - 2(y_4 - y_2)z_4,\\
y_3' &= y_7 + 2(y_3 - y_1)z_4,\\
y_4' &= y_8 + 2(y_4 - y_2)z_4,\\
M_1y_5' &= 2y_1z_1 - 2(y_3 - y_1)z_2,\\
M_1y_6' &= 2y_2z_1 - 2(y_4 - y_2)z_2 - g, \qquad t \in [0,1],\\
M_2y_7' &= 2(y_3 - y_1)z_2,\\
M_2y_8' &= 2(y_4 - y_2)z_2 - g,\\
0 &= y_1y_5 + y_2y_6,\\
0 &= (y_3 - y_1)(y_7 - y_5) + (y_4 - y_2)(y_8 - y_6),\\
0 &= y_1^2 + y_2^2 - L_1,\\
0 &= (y_3 - y_1)^2 + (y_4 - y_2)^2 - L_2.
\end{aligned} \tag{35}$$

Equations (35) correspond to the double pendulum as index-2 DAE in GGL formulation,

$$\begin{aligned}
y' &= v + G^t\nu,\\
Mv' &= f + G^t\lambda,\\
0 &= G(y)v,\\
0 &= g(y).
\end{aligned}$$

We have taken $M_1 = M_2 = 1$ and $L_1 = L_2 = 1$. We have integrated this problem with different consistent initial conditions at $t = 0$. The used criteria to stop the iterations is also (34). The results have been summarized in Tables 5 and 6. We show again the number of iterations per stage for the methods considered in Sections 3.2 and 3.3. It can be seen that the optimal predictor is more efficient than the trivial one.

Table 5. Numerical results for Problem 2 with initial conditions $(1, 0, 2, 0, 0, 0, 0, 0)$.

	Stepsize	$(\mathrm{Tol}_y, \mathrm{Tol}_z)$	Iter./Stage Optimal	Iter./Stage Trivial
Four-stages method (29)	0.02	$(10^{-7}, 10^{-3})$	4.47	6.35
	0.01	$(10^{-7}, 10^{-3})$	3.16	5.32
	0.005	$(10^{-7}, 10^{-4})$	2.23	4.36
	0.0025	$(10^{-7}, 10^{-4})$	2.00	3.92
	0.001	$(10^{-7}, 10^{-6})$	1.27	3.93
Six-stages method (32)	0.02	$(10^{-7}, 10^{-3})$	3.99	5.61
	0.01	$(10^{-7}, 10^{-3})$	2.93	4.68
	0.005	$(10^{-7}, 10^{-4})$	2.01	4.01
	0.0025	$(10^{-7}, 10^{-4})$	1.46	3.50
	0.001	$(10^{-7}, 10^{-6})$	2.00	3.73

Table 6. Numerical results for Problem 2 with initial conditions $(1, 0, 1, 1, 0, 0, 0, 0)$.

	Stepsize	$(\mathrm{Tol}_y, \mathrm{Tol}_z)$	Iter./Stage Optimal	Iter./Stage Trivial
Four-stages method (29)	0.02	$(10^{-7}, 10^{-3})$	5.17	6.45
	0.01	$(10^{-7}, 10^{-3})$	3.87	5.65
	0.005	$(10^{-7}, 10^{-4})$	2.92	4.87
	0.0025	$(10^{-7}, 10^{-4})$	2.17	4.01
	0.001	$(10^{-7}, 10^{-6})$	1.77	4.11
Six-stages method (32)	0.02	$(10^{-7}, 10^{-3})$	4.42	5.88
	0.01	$(10^{-7}, 10^{-3})$	3.39	5.05
	0.005	$(10^{-7}, 10^{-4})$	2.51	4.34
	0.0025	$(10^{-7}, 10^{-4})$	1.97	3.76
	0.001	$(10^{-7}, 10^{-6})$	2.11	3.92

6. CONCLUSIONS

In this paper, we have got predictors for a class of Runge-Kutta methods with singular matrix in the index-2 case. The numerical experiments show that the behavior of the predictor gets better when the order gets bigger.

The numerical experiments have been done with fixed step. In the general case of variable step, we have to take care with high values of the step ratio r. There are not great restrictions for moderate values of r, but for large values of r, which correspond to great increases of the stepsize h, there could be severe restrictions in the integration step for some type of problems. In such cases it could be more suitable to use a predictor of lower order but with better stability properties. Therefore it may be interesting to consider optimum predictors taking into account both aspects, order and stability. This will be worked in a future paper.

REFERENCES

1. U. Ascher and L. Petzold, Projected implicit Runge-Kutta methods for differential-algebraic equations, *SIAM J. Numer. Anal.* **28**, 1097–1120, (1991).
2. H. Bijl, M. Carpenter, V. Vatsa and C. Kennedy, Implicit time integration schemes for the unsteady compressible Navier-Stokes equations: Laminar flow, *J. Comput. Phys.* **179**, 313–329, (2002).
3. K. Burrage, F. Chipman and P. Muir, Order results for mono-implicit Runge-Kutta methods, *SIAM J. Numer. Anal.* **31** (3), 876–891, (1994).
4. M. Calvo, P. Laburta and J.I. Montijano, Starting algorithms for gauss Runge-Kutta methods for Hamiltonian systems, *Computers Math. Applic.* **45** (1–3), 401–410, (2003).
5. M. Calvo, P. Laburta and J.I. Montijano, Two-step high order starting values for implicit Runge-Kutta methods, *Adv. Comput. Math.* **19** (4), 401–412, (2003).

6. M.P. Calvo, High-order starting iterates for implicit Runge-Kutta methods. An improvement for variable-step symplectic integrators, *IMA Journal of Numerical Analysis* **22** (1), 153–166, (2002).
7. F. Cameron, A class of low order DIRK methods for a class of DAEs, *Applied Numerical Mathematics* **31**, 1–16, (1999).
8. J. Cash, A class of implicit Runge-Kutta methods for the numerical integration of stiff ordinary differential equations, *J. ACM* **22** (4), 504–511, (1975).
9. S. González-Pinto, J.I. Montijano and S.P. Rodríguez, On the starting algorithms for fully implicit Runge-Kutta methods, *BIT* **40** (4), 685–714, (2000).
10. S. González-Pinto, J.I. Montijano and S.P. Rodríguez, Stabilized starting algorithms for collocation Runge-Kutta methods, *Computers Math. Applic.* **45** (1–3), 411–428, (2003).
11. E. Hairer, C. Lubich and M. Roche, *The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods, Lecture Notes in Mathematics, Volume 1409*, Springer, (1989).
12. E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems*, Springer, Berlin, (1991).
13. I. Higueras and T. Roldán, Starting algorithms for some DIRK methods, *Numerical Algorithms* **23**, 357–369, (2000).
14. I. Higueras and T. Roldán, IRK methods for index 2 and 3 DAEs: Starting algorithms, *BIT* **43-1**, 67–92, (2003).
15. L. Jay, Convergence of a class of Runge-Kutta methods for differential-algebraic systems of index 2, *BIT* **33**, 137–150, (1993).
16. L. Jay, Iterative solution of SPARK methods applied to DAEs, *Numerical Algorithms* **31**, 171–191, (2002).
17. L. Jay, Solution of index 2 implicit differential-algebraic equations by Lobatto Runge-Kutta methods, *BIT* **43** (1), 93–106, (2003).
18. C.A. Kennedy and M.H. Carpenter, Additive Runge-Kutta schemes for convection-diffusion-reaction equations, *Applied Numerical Mathematics* **44** (1–2), 139–181, (2003).
19. M. Laburta, Starting algorithms for IRK methods, *J. Comput. Appl. Math.* **83**, 269–288, (1997).
20. C. Lubich, On projected Runge-Kutta methods for differential-algebraic equations, *BIT* **31**, 545–550, (1991).
21. T. Roldán, Implicit Runge-Kutta methods for DAEs: Starting algorithms, Ph.D. Thesis, Departamento de Matemática e Informática, Universidad Pública de Navarra, Spain, (2000).
22. T. Roldán and I. Higueras, IRK methods for DAE: Starting algorithms, *J. Comput. Appl. Math.* **111**, 77–92, (1999).
23. R. Williams, K. Burrage, I. Cameron and M. Kerr, A four-stage index 2 diagonally implicit Runge-Kutta method, *Applied Numerical Mathematics* **40-3**, 415–432, (2002).